Fig. 1

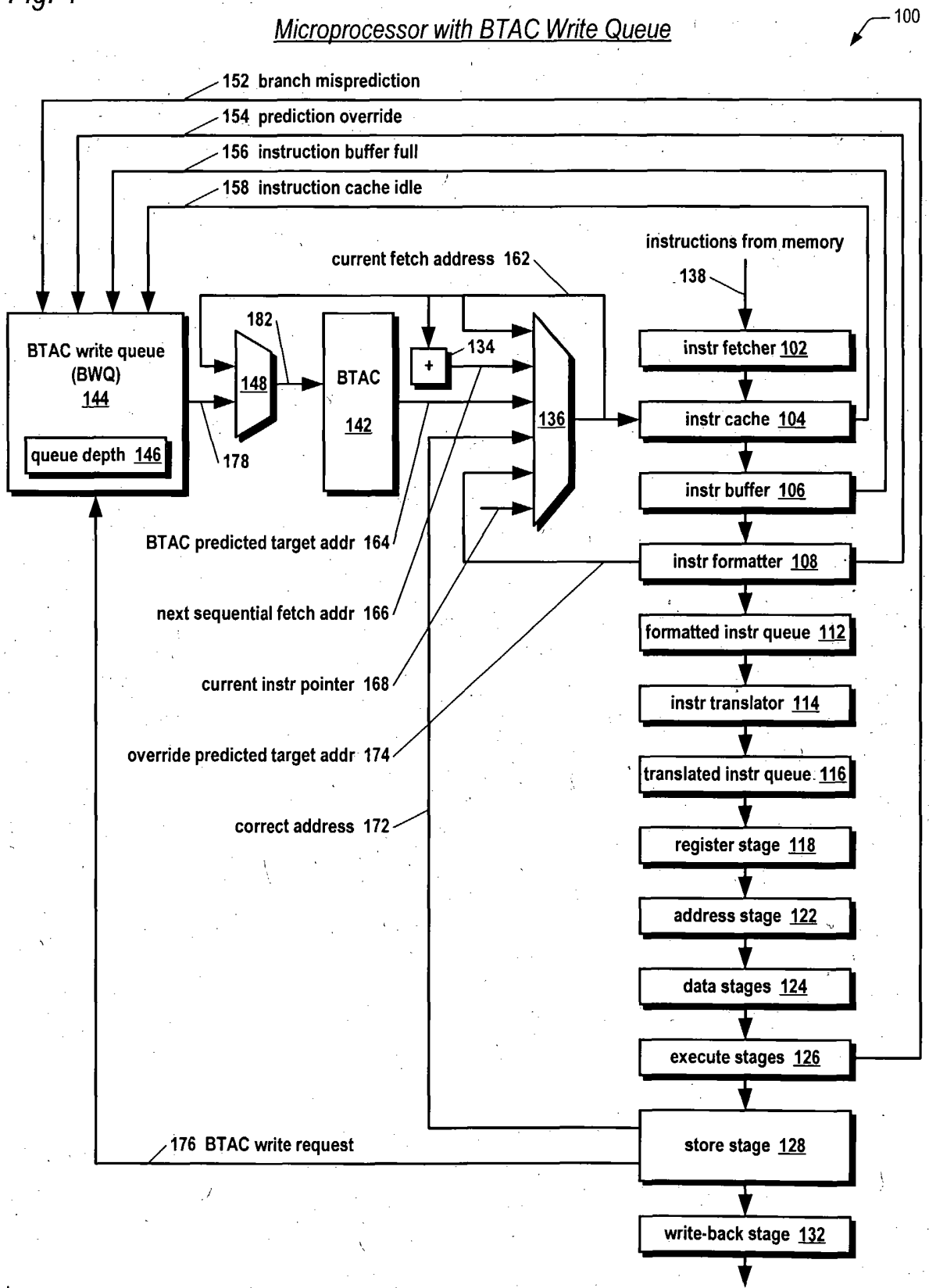

Microprocessor with BTAC Write Queue
100
152 branch misprediction
154 prediction override
156 instruction buffer full
158 instruction cache idle
current fetch address 162
instructions from memory
138
BTAC write queue (BWQ) 144
queue depth 146
182
148
178
BTAC 142
+
134
136
BTAC predicted target addr 164
next sequential fetch addr 166
current instr pointer 168
override predicted target addr 174
correct address 172
176 BTAC write request
instr fetcher 102
instr cache 104
instr buffer 106
instr formatter 108
formatted instr queue 112
instr translator 114
translated instr queue 116
register stage 118
address stage 122
data stages 124
execute stages 126
store stage 128
write-back stage 132

Fig. 2

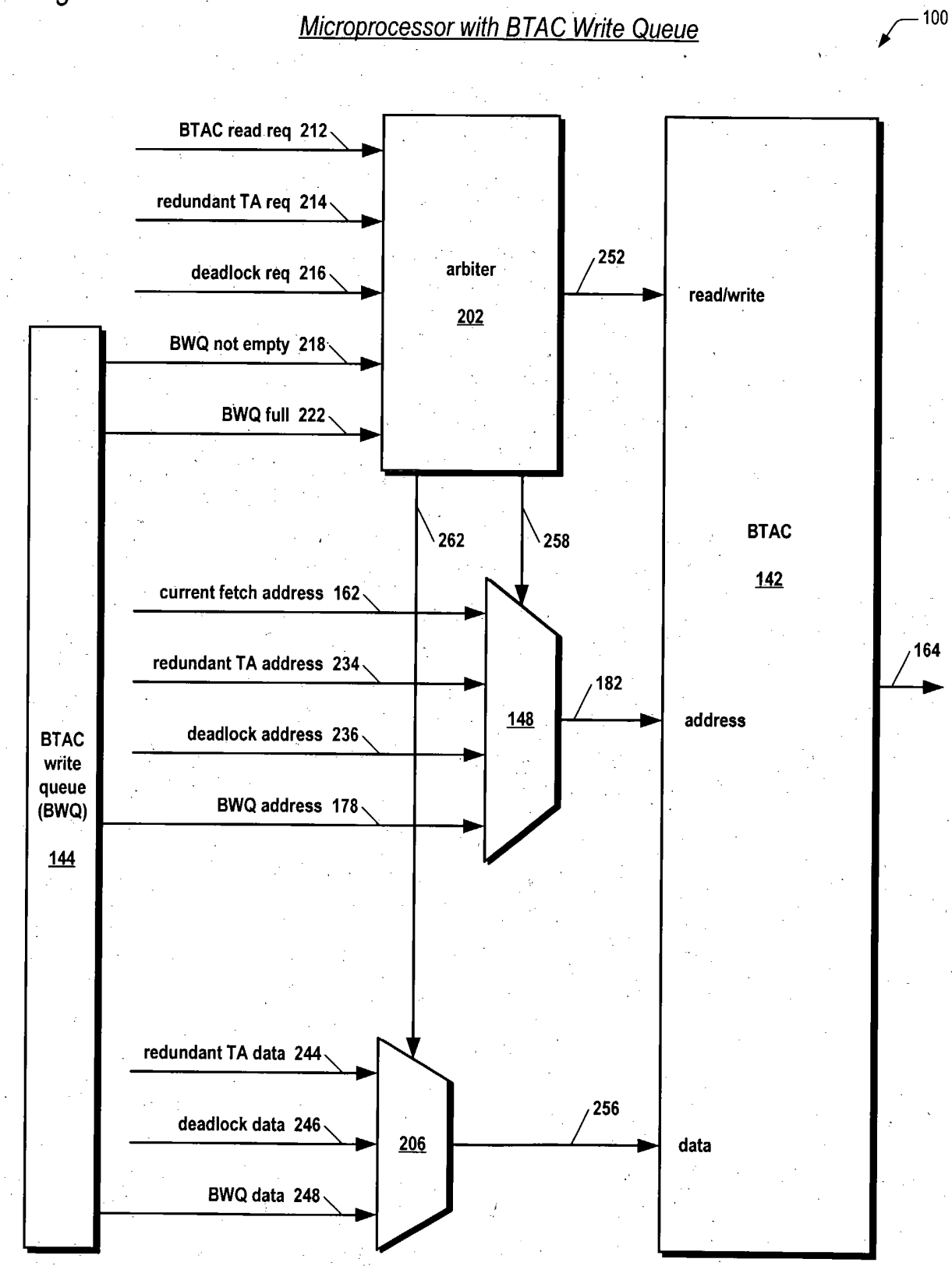

Microprocessor with BTAC Write Queue
100
BTAC read req 212
redundant TA req 214
deadlock req 216
BWQ not empty 218
BWQ full 222
arbiter
202
252
read/write
BTAC
142
262
258
current fetch address 162
redundant TA address 234
deadlock address 236
BWQ address 178
148
182
address
164
BTAC write queue (BWQ)
144
redundant TA data 244
deadlock data 246
BWQ data 248
206
256
data

Fig. 3

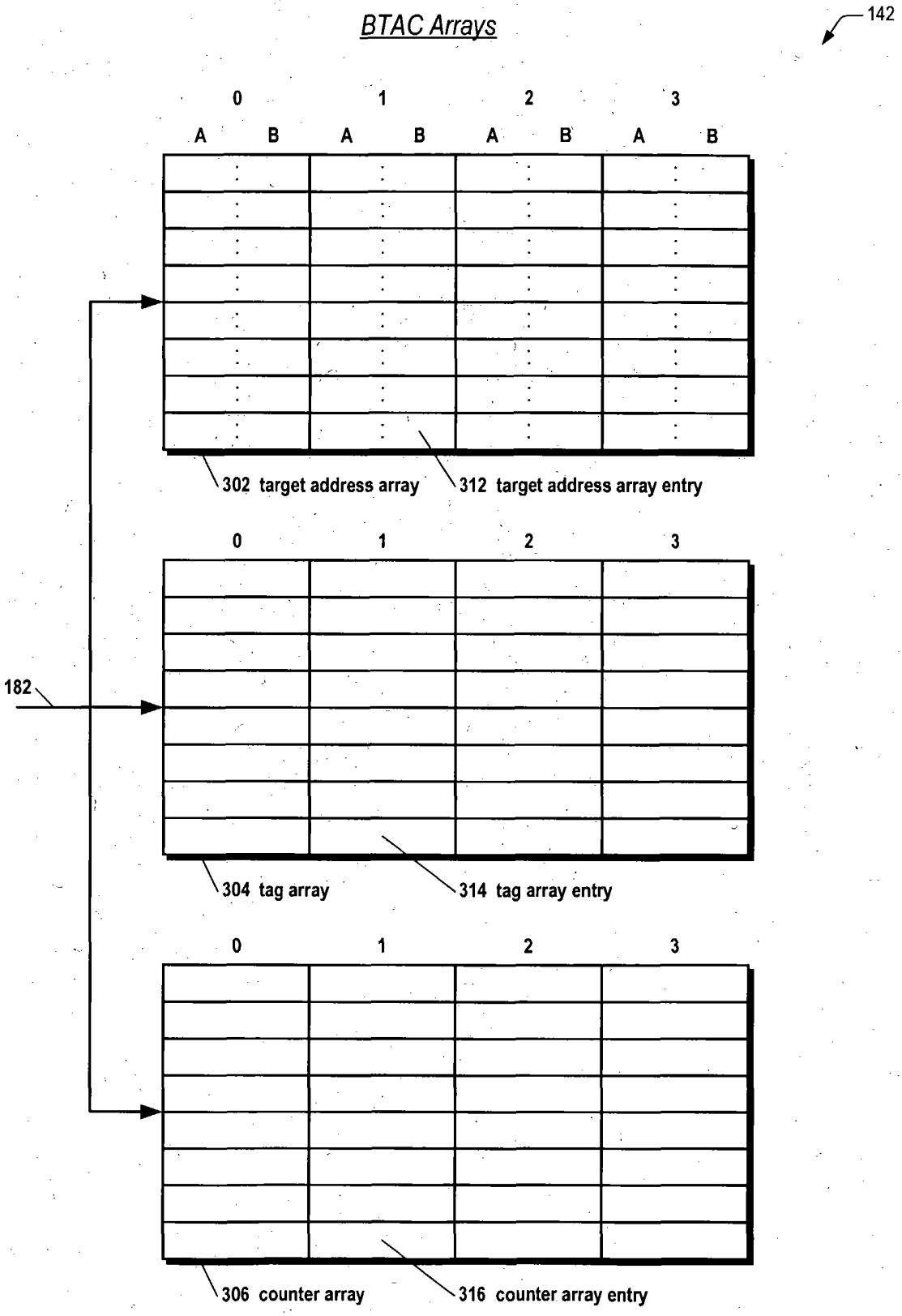

BTAC Arrays
142
0
1
2
3
A
B
A
B
A
B
A
B
302 target address array
312 target address array entry
182
304 tag array
314 tag array entry
306 counter array
316 counter array entry

Fig. 4

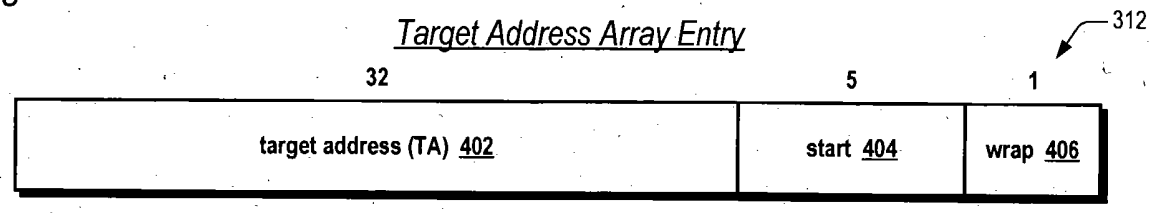


Fig. 5

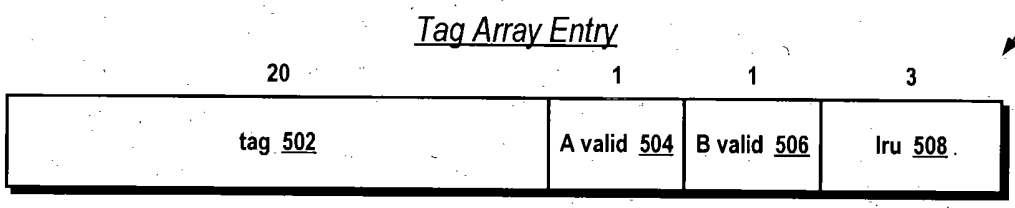


Fig. 6

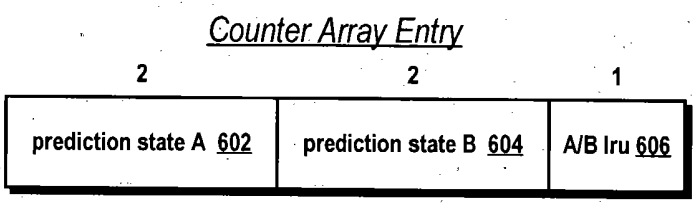


Fig. 7

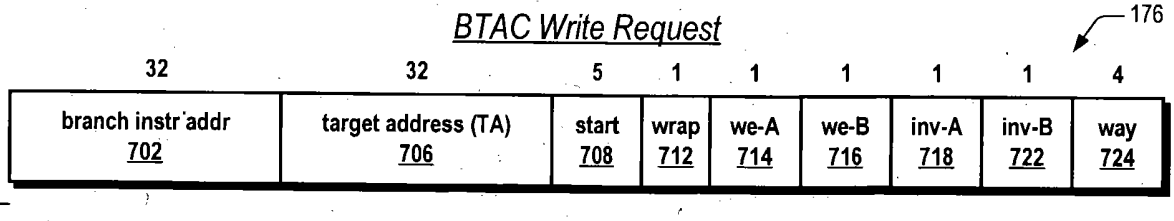


Fig. 8

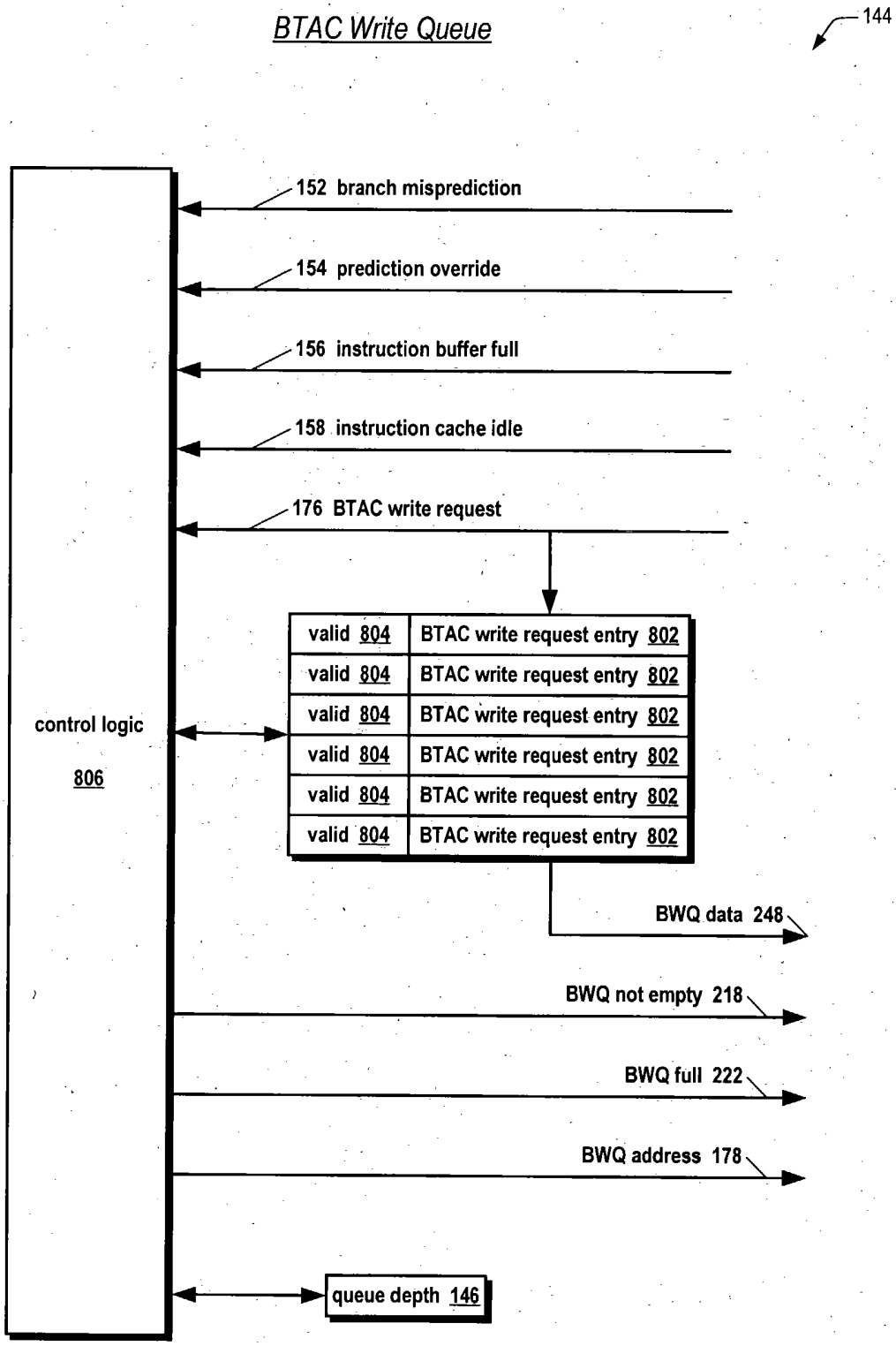

BTAC Write Queue
144
152 branch misprediction
154 prediction override
156 instruction buffer full
158 instruction cache idle
176 BTAC write request
control logic
806
valid 804
BTAC write request entry 802
BWQ data 248
BWQ not empty 218
BWQ full 222
BWQ address 178
queue depth 146

Fig. 9

BTAC Write Queue Operation

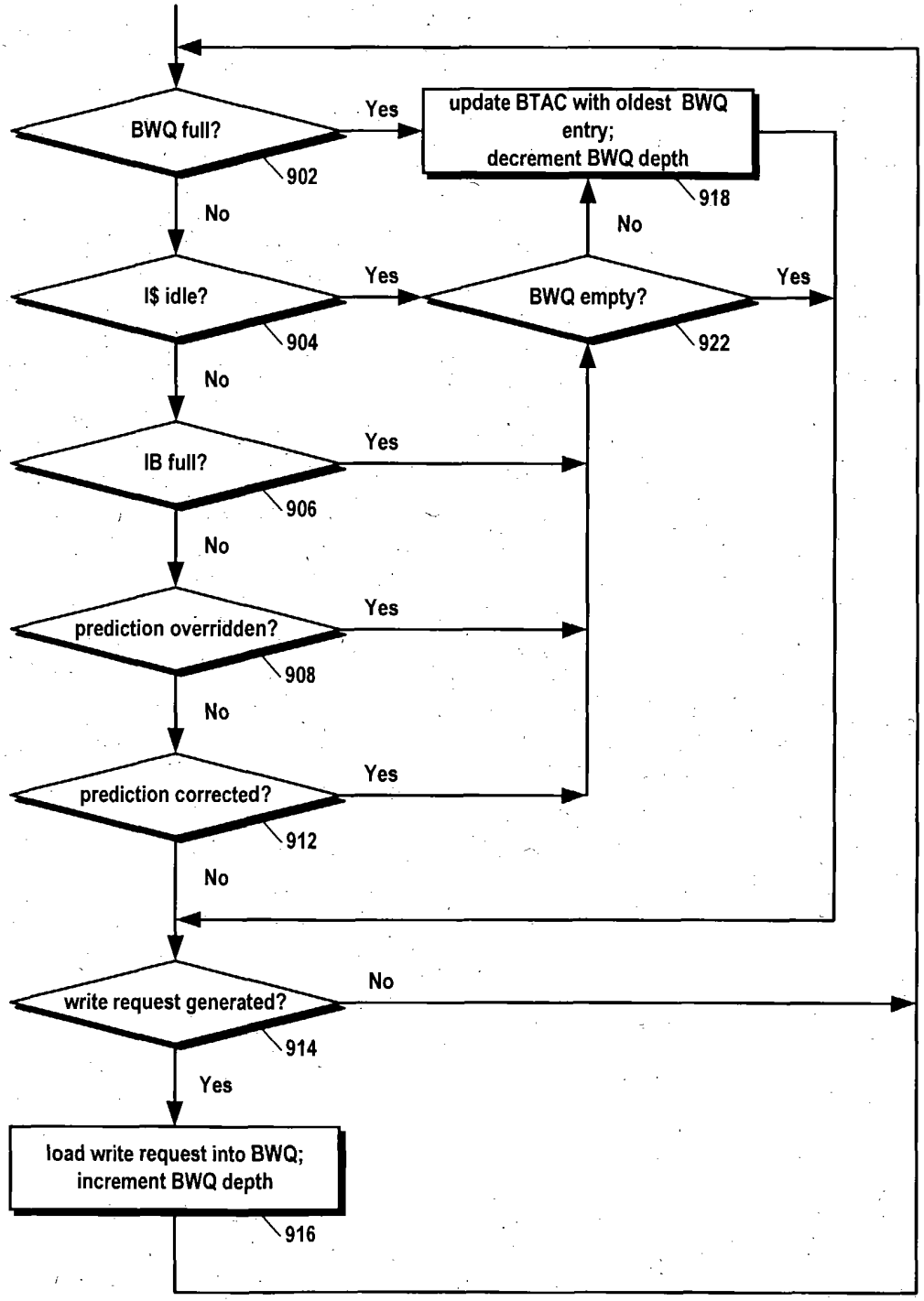


Fig. 10

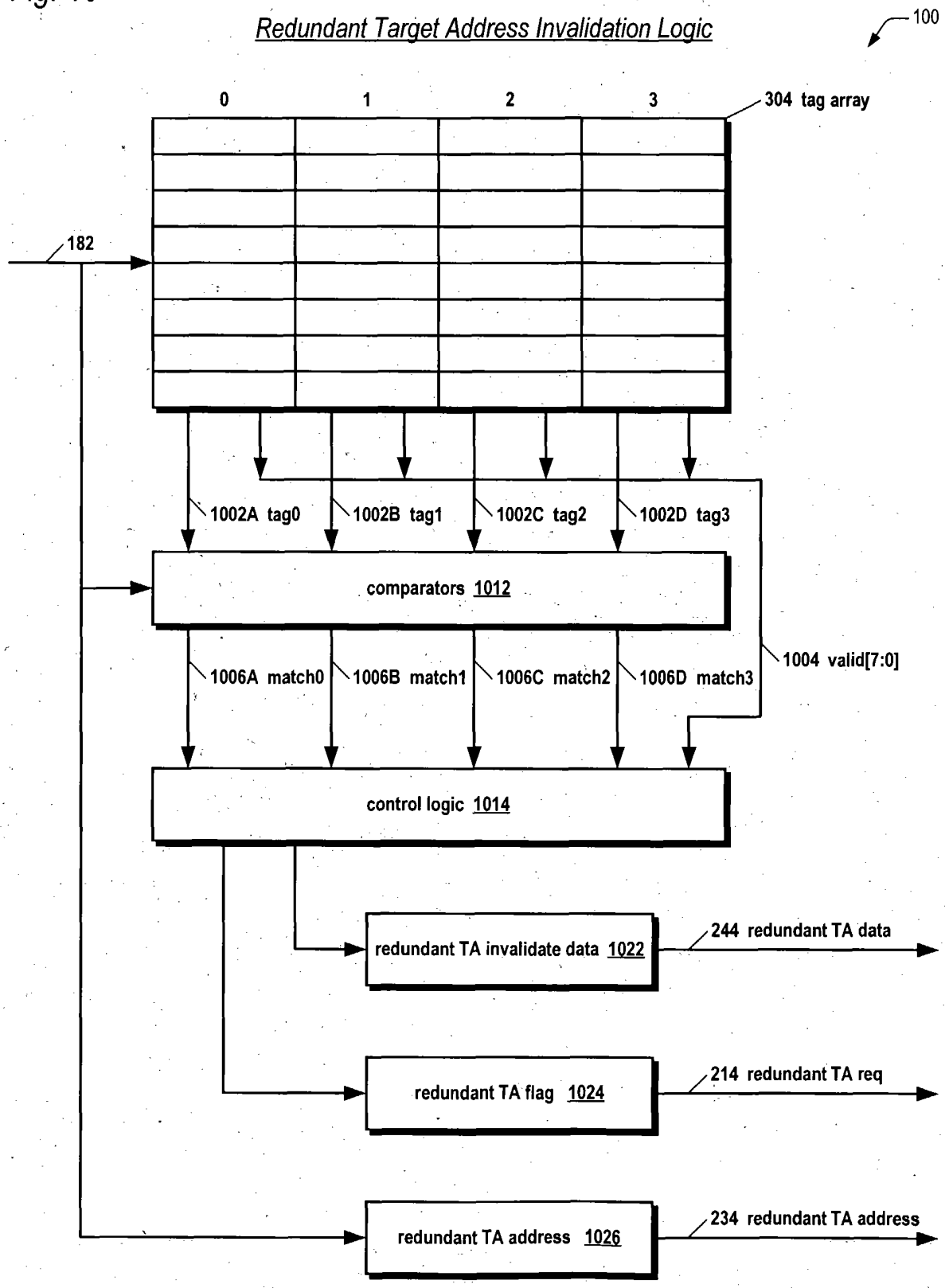

Redundant Target Address Invalidation Logic
100
0
1
2
3
304 tag array
182
1002A tag0
1002B tag1
1002C tag2
1002D tag3
comparators 1012
1004 valid[7:0]
1006A match0
1006B match1
1006C match2
1006D match3
control logic 1014
redundant TA invalidate data 1022
244 redundant TA data
redundant TA flag 1024
214 redundant TA req
redundant TA address 1026
234 redundant TA address

Fig. 11

Redundant Target Address Invalidation Operation

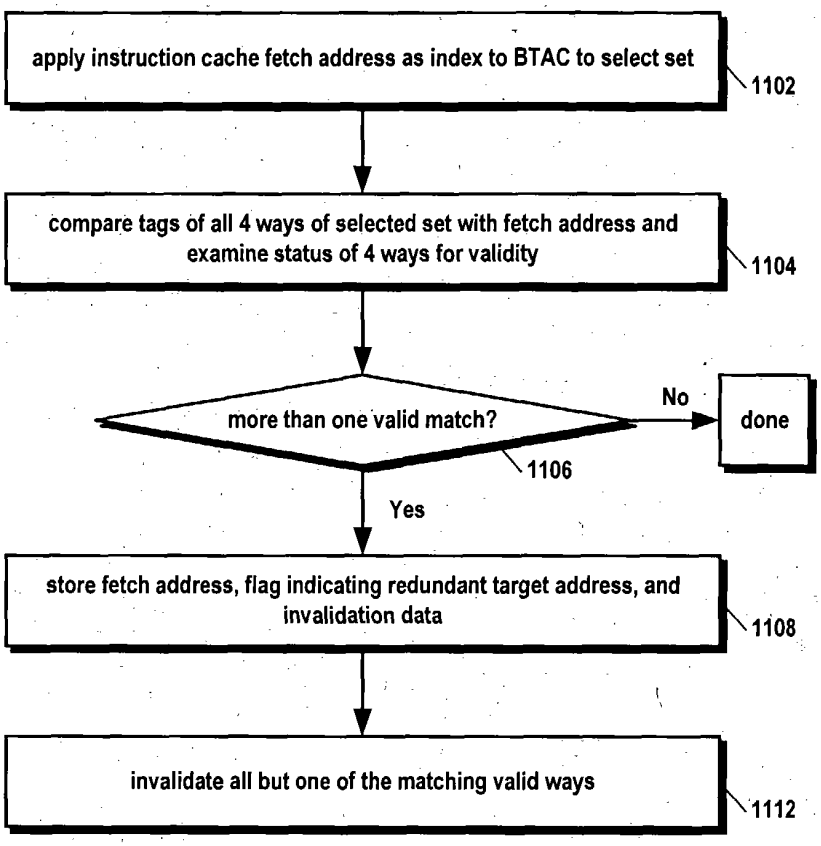


Fig. 12

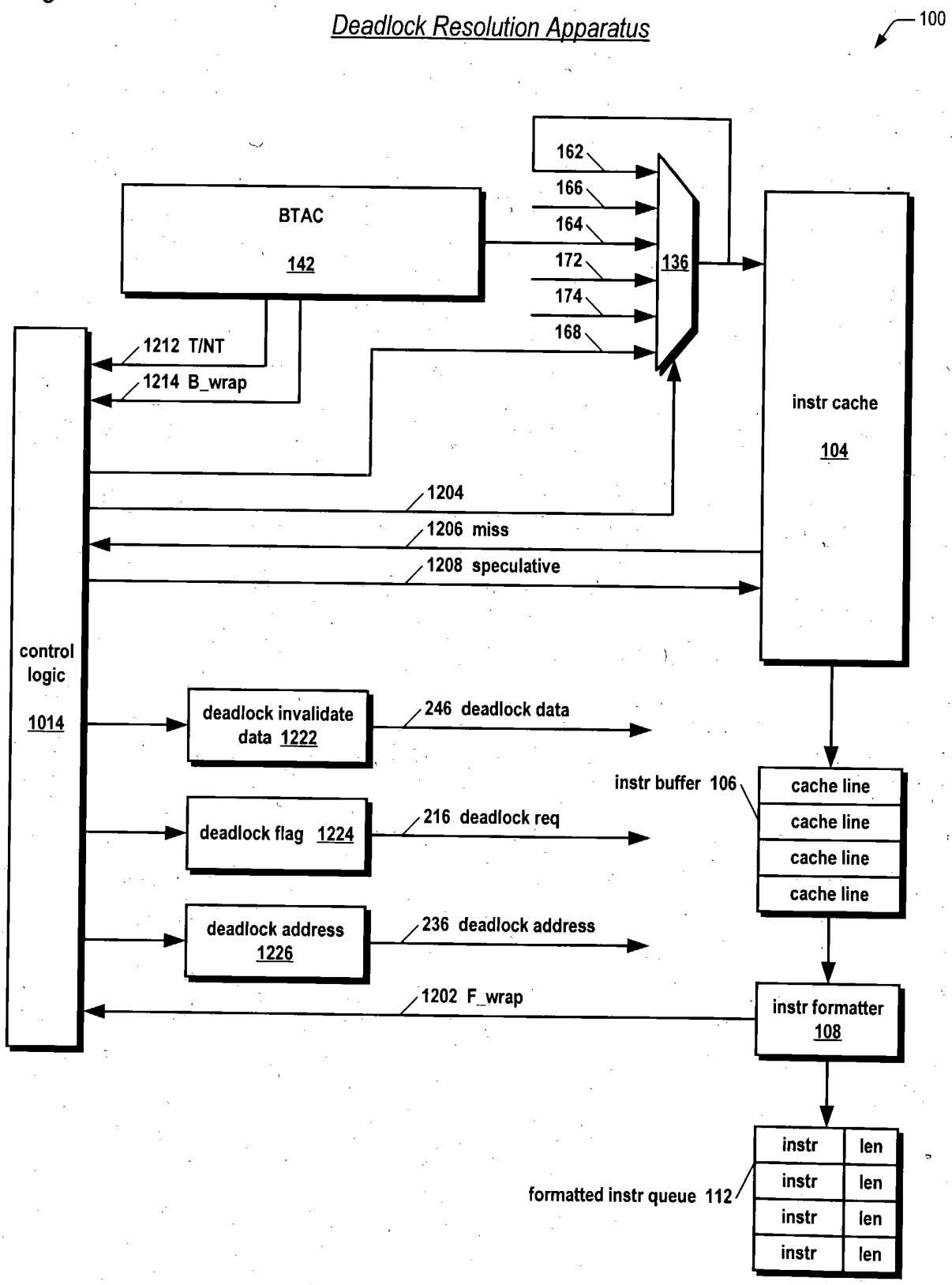
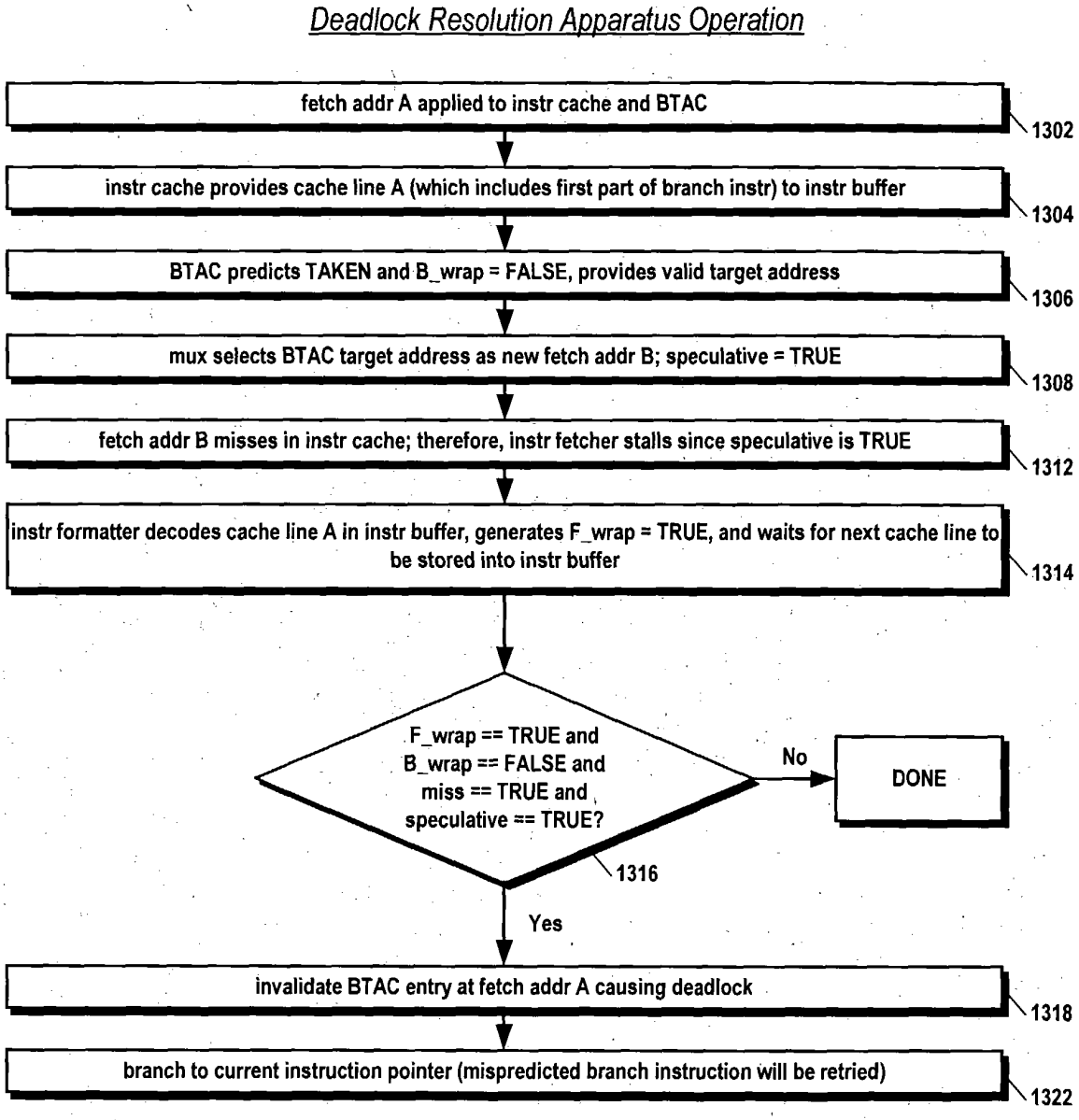

Deadlock Resolution Apparatus
100
BTAC
142
162
166
164
172
174
168
136
instr cache
104
1212 T/NT
1214 B_wrap
1204
1206 miss
1208 speculative
control logic
1014
deadlock invalidate data 1222
246 deadlock data
instr buffer 106
cache line
cache line
cache line
cache line
deadlock flag 1224
216 deadlock req
deadlock address 1226
236 deadlock address
1202 F_wrap
instr formatter 108
instr
len
instr
len
instr
len
instr
len
formatted instr queue 112

Fig. 13


Deadlock Resolution Apparatus Operation
fetch addr A applied to instr cache and BTAC
1302
instr cache provides cache line A (which includes first part of branch instr) to instr buffer
1304
BTAC predicts TAKEN and B_wrap = FALSE, provides valid target address
1306
mux selects BTAC target address as new fetch addr B; speculative = TRUE
1308
fetch addr B misses in instr cache; therefore, instr fetcher stalls since speculative is TRUE
1312
instr formatter decodes cache line A in instr buffer, generates F_wrap = TRUE, and waits for next cache line to be stored into instr buffer
1314
F_wrap == TRUE and B_wrap == FALSE and miss == TRUE and speculative == TRUE?
1316
No
DONE
Yes
invalidate BTAC entry at fetch addr A causing deadlock
1318
branch to current instruction pointer (mispredicted branch instruction will be retried)
1322